#8

MAT-V07839 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: T. Nishimura et al. : Art Unit:
Serial No.: 09/403,560 : Examiner:
Filed: January 12, 2000
FOR: DATA TRANSFER METHOD

VERIFICATION OF A TRANSLATION

Assistant Commissioner for Patents
Washington, D.C. 20231
SIR :

I, the below named translator, hereby declare that:

1. My name and post office address are as stated below.

2. That I am knowledgeable in the English language and in the language of JPH09-106995, and I believe the attached English translation to be a true and complete translation of JPH09-106995.

3. The document for which the attached English translation is being submitted is a patent application on an invention entitled DATA TRANSFER METHOD.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date: April 26, 2004

Masakazu Teraoka

Full name of the Translator

M. Teraoka

Signature of the Translator

Matsushita Technical Information Services Co., Ltd.,
Osaka Business Park, Matsushita IMP Bldg., 18th Floor,
1-3-7, Shiromi, Chuo-ku, Osaka-shi, Osaka, 540-6318 Japan

Post Office Address

[NAME OF THE DOCUMENT] Patent Application
[ARRANGEMENT NUMBER] 2054091207
[DATE OF FILING] April 24, 1997
[ADDRESS] Director-General of the Patent Office
[INTERNATIONAL PATENT CLASSIFICATION] G06F 13/00
[TITLE OF THE INVENTION] Data Transfer Method
[NUMBER OF CLAIMS] 13

[INVENTORS]
[NAME] Takuya NISHIMURA
[ADDRESS] c/o Matsushita Electric Industrial Co., Ltd.
1006, Oaza-Kadoma, Kadoma-shi, Osaka-fu
[NAME] Hiroyuki IITSUKA
[ADDRESS] c/o Matsushita Electric Industrial Co., Ltd.
1006, Oaza-Kadoma, Kadoma-shi, Osaka-fu
[NAME] Masazumi YAMADA
[ADDRESS] c/o Matsushita Electric Industrial Co., Ltd.
1006, Oaza-Kadoma, Kadoma-shi, Osaka-fu

[APPLICANT]
[INDENTIFICATION NUMBER] 000005821
[NAME] Matsushita Electric Industrial Co., Ltd.
[ADDRESS] 1006, Oaza-Kadoma, Kadoma-shi, Osaka-fu
[POSTAL CODE] 571-8501

[AGENT]
[IDENTIFICATION NUMBER] 100078204
[NAME] Tomoyuki TAKIMOTO, Patent Attorney
[ADDRESS] c/o Matsushita Electric Industrial Co., Ltd.
1006, Oaza-Kadoma, Kadoma-shi, Osaka-fu
[POSTAL CODE] 571-8501

[SELECTED AGENT]

[IDENTIFICATION NUMBER] 100097445

[NAME] Fumio IWAHASHI, Patent Attorney

[ADDRESS] c/o Matsushita Electric Industrial Co., Ltd.
1006, Oaza-Kadoma, Kadoma-shi, Osaka-fu

[POSTAL CODE] 571-8501

[REPRESENTATION OF FEE]

[PAYING METHOD] In-advance payment

[NUMBER IN LEDGER OF IN-ADVANCE PAYMENT] 011305

[AMOUNT] 21000

[LIST OF ARTICLES FILED]

[NAME OF ARTICLE] Specification 1

[NAME OF ARTICLE] Drawing 1

[NAME OF ARTICLE] Abstract 1

[NUMBER OF GENERAL POWER] 9702380

[NECESSITY OF PROOF] Necessary

[NAME OF THE DOCUMENT] Specification

[TITLE OF THE INVENTION]

Data Transfer Method

[SCOPE OF THE CLAIMS]

[CLAIM 1]

A method for transferring data on a bus system in which both synchronous communication and asynchronous communication are employed;

said synchronous communication is for any device on the bus to receive synchronous data;

said asynchronous communication is for a predetermined device to receive asynchronous data;

said synchronous data may contain actual data; and said synchronous data also contains encryption identification information at an area other than said actual data; said encryption identification information indicates the status of encryption of said actual data; and encrypted actual data is decrypted using decrypting information obtained through the following steps:

a) a receiving device receiving said synchronous data makes a request for decrypting information of said actual data to a sending device sending said synchronous data via said asynchronous communication, if said encryption identification information indicates that said actual data is encrypted;

b) said sending device receiving said request sends one of:

i) encrypted decrypting information of said actual data; and

ii) data required for obtaining said decrypting information for obtaining said decrypting information,

to said receiving device via said asynchronous communication; and

c) said receiving device executes one of:

i) taking out said decrypting information from said encrypted decrypting information when said receiving device receives said encrypted decrypting information; and

ii) obtaining said decrypting information using said data for obtaining said decrypting information when said receiving device receives said data for obtaining decrypting information.

[CLAIM 2]

The method for transferring data as defined in claim 1, wherein a plurality of types of procedures are available between the steps of detecting encryption of said actual data and obtaining said decrypting information by said receiving device receiving said synchronous data; and said receiving device executes the next steps for obtaining said decrypting information before requesting said decrypting information:

i) querying said sending device of types of procedures executable by said sending device before requesting said decrypting information;

ii) selecting a procedure from those executable by both sending device and receiving device; and

iii) obtaining said decrypting information in accordance with said selected procedure.

[CLAIM 3]

The method for transferring data as defined in claim 2, wherein said asynchronous data transmitted between said sending device and said receiving device in accordance with said selected procedure contains an identifier for indicating the type of said procedure executed.

[CLAIM 4]

The method for transferring data as defined in claim 1,2,or 3, wherein said receiving device authenticates whether said sending device is an authorized sending device before making a request for said decrypting information.

[CLAIM 5]

The method for transferring data as defined in claim 1, 2, or 3, wherein said sending device receiving a request for said decrypting information authenticates that said receiving device is an authorized receiving device before sending encrypted decrypting information of said actual data after confirming.

[CLAIM 6]

The method for transferring data as defined in claims 1,2, or 3, wherein said sending device and said receiving device mutually authenticate that both are authorized sending device and receiving device before said receiving device makes a request for said decrypting information.

[CLAIM 7]

The method for transferring data as defined in one of claims 1 through 6, wherein the next steps are executed before said receiving device makes a request for said decrypting information:

i) said receiving device sends information required by said sending device at least for creating a common key to said sending device; and

ii) said sending device sends information required by said receiving device at least for creating said common key to said receiving device; and then said sending device encrypts said decrypting information using said common key and sends said encrypted decrypting information; and said receiving device takes out said decrypting information from said encrypted decrypting information received using said common key.

[CLAIM 8]

The method for transferring data as defined in claim 1,2, or 3, wherein only said actual data is encrypted.

[CLAIM 9]

The method for transferring data as defined in claim 1, 2, or 3, wherein said sending device has a signal source of said actual data inside and determines encryption of each of said actual data in a fixed length unit outputted from said signal source; and said sending device places encrypted actual data and non-encrypted actual data in different output units of said synchronous communication, and then outputs them to said bus system.

[CLAIM 10]

The method for transferring data as defined in claim 9, wherein said sending device specifies proportion of encrypted actual data and non-encrypted actual data to said receiving device by using asynchronous communication, and said sending device changes proportion of encryption according to said specification.

[CLAIM 11]

The method for transferring data as defined in claim 1, 2, or 3, wherein said sending device has a signal source of said actual data inside and determines the proportion of encrypting said actual data in a fixed length unit outputted from said signal source; and said sending device places said actual data in output units of said synchronous communication, and then outputs them to said bus system.

[CLAIM 12]

The method for transferring data as defined in claim 11, wherein said sending device specifies proportion of encryption to said receiving device by using asynchronous communication, and said sending device changes proportion of encryption according to said specification.

[CLAIM 13]

The method for transferring data as defined in claim 1, 2, or 3, wherein said sending device sends synchronous data without including actual data at least before being requested for decrypting information, and start to send said synchronous data including said actual data at least after receiving a request for said decrypting information.

[DETAILED DESCRIPTION OF THE INVENTION]

[0001]

[FIELD OF THE INVENTION]

The present invention relates to data transfer methods between devices sending and receiving digital data.

[0002]

[PRIOR ART]

One conventional data transfer method adopts the IEEE1394 standard (IEEE: The Institute of Electrical and Electronics Engineers, Inc.). (Reference: IEEE Std 1394: 1995, High Performance Serial Bus.) In data transfer specified by the IEEE 1394 standard, there are two methods of communication. One is isochronous communication, which is suitable for transferring synchronous data such as digital video signals and digital audio signals. The other is asynchronous communication, which is suitable for transferring asynchronous data such as control signals. Both methods of communication are applicable on the IEEE 1394 bus network. Isochronous communication is what is called broadcast communication, and an isochronous packet output from one device coupled to the IEEE 1394 bus is receivable by all the other devices coupled to the same bus. On the other hand, asynchronous communication is applicable to both one-to-one communication and broadcast communication. Each asynchronous packet from one device coupled to the bus contains an identifier specifying the device(s) to which that packet is addressed. If this identifier specifies a particular device, only the device specified by the identifier receives the asynchronous packet. If the identifier specifies broadcast, all the devices coupled to the same bus receive the asynchronous packet.

[0003]

At present, the IEC (International Electrotechnical Commission) prepares to stipulate the IEC1883 standard (hereafter referred to as AV protocol) for transferring digital audio signals and digital video signals or transmitting data between devices coupled to an IEEE 1394 bus, employing the data transfer method conforming to the IEEE 1394 standard. In the AV protocol, video and audio data is located in the isochronous packet as shown in Fig. 5 and transferred. The isochronous packet includes a CIP (Common Isochronous Packet) header. The CIP header carries information that includes the type of AV data, the identification number of the device which is sending the isochronous packet, and the like.

[0004]

Fig. 5 shows the format of the isochronous packet used in the AV protocol. The isochronous packet comprises an isochronous packet header 900, header CRC 901, isochronous payload 902, and data CRC 903. The isochronous packet header 900 contains a tag 907. The tag 907 shows that the isochronous packet conforms to the AV protocol when its value is 1. When the value of the tag 907 is 1, which means that the isochronous packet conforms to the AV protocol, the isochronous payload 902 has a CIP header 904 at its beginning. The CIP header 904 comprises a source ID 906 that identifies the device transmitting the isochronous packet. The CIP header 904 also comprises FMT 908 and FDF 909 that specify the type of actual data 905 in the isochronous payload 902. Digital AV data is contained in the actual data 905, but the actual data 905 is not always contained in the isochronous payload 902. Some packets may have an isochronous payload 902 that contains only the CIP header 904 without the actual data 905.

[0005]

There is a group of commands called the AV/C Command Set for controlling devices in accordance with the AV protocol (Reference: 1394 TRADE ASSOCIATION Specification for AV/C Digital Interface Command Set Version 1.0, September 13, 1996). These commands and their responses are transferred by means of asynchronous communication.

[0006]

[PROBLEMS TO BE SOLVED BY THE INVENTION]

In the conventional data transfer method as described above, compatibility with conventional devices that are not designed for transferring an encrypted isochronous payload 902 cannot be secured when an encrypted isochronous packet, that contains the isochronous payload 902 that has been encrypted for copyright protection, is sent. More specifically, conventional devices are designed with the precondition that the CIP header 904 is normally positioned at the beginning of the isochronous payload 902. Accordingly, if the isochronous payload 902 is encrypted, conventional devices cannot correctly read out the encrypted CIP header 904, and decide that the isochronous packet does not conform to the AV protocol. A device receiving encrypted isochronous packets thus may not operate properly. In other words, such receiving devices cannot determine the type of data contained in the actual data 905, resulting in an inability to identify the device transmitting the isochronous packet. In addition, asynchronous communication such as questions to the sending device is disabled. Accordingly, receiving operations cannot be normally carried out.

[0007]

Furthermore, if the isochronous packet output from the sending device is encrypted while the receiving device is receiving the data, some conventional devices may not be able to correctly read out the CIP header 904 as soon as encryption starts, resulting in inability to receive data properly.

[0008]

In order to send AV information encrypted for copyright protection from the sending device and decrypt the encrypted AV data by the authorized receiving device, the sending device needs to give decrypting information for decryption to the authorized receiving device. In the conventional data transfer method, however, the sending device may be required to execute extremely complicated procedures in order to specify the receiving device. More specifically, each isochronous packet contains the source ID 906 that is the identifier of the sending device, but these packets do not contain information that identifies which device is authorized to receive these packets. The sending device thus cannot check which device is receiving the isochronous packets during transmission of the isochronous packets. In order to find which of the devices coupled to the IEEE 1394 bus is receiving the data, the sending device may require to query the data receiving status of every device coupled to the same bus. This makes the procedures for giving key information for decryption extremely complicated.

[0009]

A data transfer method of the present invention satisfies the conventional communication standard even in the case of sending encrypted video and audio information via isochronous communication. In addition, the present invention offers a data transfer method for preventing erroneous operation even if conventional receiving devices receive isochronous packets containing encrypted video and audio data.

[0010]

The present invention still further offers a data transfer method that significantly simplifies procedures for giving key information for decryption from a sending device to an authorized receiving device.

[0011]

[MEANS TO SOLVE THE PROBLEMS]

In a data transfer method of the present invention, synchronous data transferred via isochronous communication contains encryption identification information that indicates encryption status of actual data and actual data, and only the actual data is encrypted.

[0012]

To solve another problem in the conventional data transfer method, the encryption identification information that indicates encryption status of the actual data in the synchronous data is sent together with the actual data from the sending device. Accordingly, receiving device can detect that the actual data is encrypted based on this encryption identification information and requests decrypting information from the sending device in the data transfer method of the present invention. Then, the receiving device receiving the decrypting information sent from the sending device upon request decrypts the actual data using this decrypting information to complete data transfer.

[0013]

[REDUCTION TO PRACTICE]

According to the present invention, a method for transferring data on a bus system uses both isochronous communication and asynchronous communication;
said isochronous communication is for any device on the bus to receive synchronous data;
said asynchronous communication is for a predetermined device to receive asynchronous data;
said synchronous data may contain actual data; said synchronous data also contains encryption identification information at an area other than said actual data; said encryption identification information indicates the status of encryption of said actual data; and encrypted actual data is decrypted using decrypting information obtained through the following steps:

a) a receiving device receiving said synchronous data makes a request for decrypting information of said actual data to a sending device sending said synchronous data via said asynchronous communication, if said encryption identification information indicates that said actual data is encrypted;

b) said sending device receiving said request sends one of:

i) encrypted decrypting information of said actual data; and

ii) data required for obtaining said decrypting

to said receiving device via said asynchronous communication; and

d) said receiving device executes one of:

i) taking out said decrypting information from said encrypted decrypting information when said receiving device receives said encrypted decrypting information; and

ii) obtaining said decrypting information using said data for obtaining said decrypting information when said receiving device receives said data for obtaining decrypting information.

[0014]

Also in the data transfer method of the present invention, the receiving device receiving synchronous data checks for the encryption identification information contained in the synchronous data. If the receiving device detects that the actual data is encrypted, the receiving device requests decrypting information for decrypting the actual data from the sending device. This request is made using asynchronous communication. At receiving this request, the sending device checks the packet header of received command to identify the device making the request, i.e., the receiving device. The sending device then gives decrypting information to the identified receiving device using a command via asynchronous communication, enabling to realize the data transfer method with extremely simple procedures for giving decrypting information from the sending device to the receiving device.

[0015]

Moreover, in the data transfer method of the present invention, only the actual data in the synchronous data is encrypted, and the encryption identification information indicating the encryption status of the actual data is included in the synchronous data. This enables to transfer the CIP header without being encrypted, preventing erroneous operation when the conventional device receives such encrypted synchronous data. In other words, the present invention realizes a data transfer method that assures compatibility with the conventional data transfer method and eliminates the possibility of erroneous operation when the conventional receiving device receives encrypted synchronous data.

[0016]

Furthermore, the data transfer method of the present invention eliminates the possibility of erroneous operation of the receiving device receiving data when encryption of synchronous data starts while continuously receiving synchronous data from the sending device because the CIP header is not encrypted and transferred as it is.

[0017]

A preferred embodiment of the present invention is described next with reference to the drawings.

Fig. 1 shows a format of the payload of an isochronous packet to be transferred in the preferred embodiment of the present invention. The preferred embodiment is one example of the transfer of a TSP (Transport Packet) in accordance with MPEG (the Moving Picture Expert Group) specifications. The encryption information (ENC) 910 indicates whether the actual data 905 is encrypted or not.

[0018]

Fig. 2 shows the relation between sending and receiving devices in the preferred embodiment of the present invention. A sending device 110 and receiving device 128 are coupled via an IEEE 1394 bus 111.

A signal source 100 outputs an MPEG transport packet TSP (not illustrated) in an 188 byte unit, that is sent via the 1394 bus 111, to an encrypter 101. In other words, in the preferred embodiment, the signal source 100 outputs data with a fixed length of 188 bytes. The encrypter 101 encrypts and outputs the TSP received from the signal source 100 using an encryption key 109 provided by a key generator 106. In the preferred embodiment, the encryption key 109 is equivalent to the decrypting information. An output command 105 is a command from the key generator 106 to the encrypter 101. There are three types of commands: normal output, encrypted output, and empty output. If the encrypter 101 receives the output command 105 for normal output, the TSP received from the signal source 100 is output without modification, and registers the value 0 as the encrypting information 910. If the output command 105 is for encrypted output, the encrypter 101 encrypts the TSP with the encryption key 109 received from the key generator 106, and registers the value 1 as the encrypting information 910.

[0019]

A source packet generator 102 adds a 4-byte source packet header to the 188-byte TSP received from the encrypter 101, and outputs a 192-byte source packet. A CIP block generator 103 adds a CIP header 954 to the source packet received from the source packet generator 102. Here, the CIP block generator 103 places the encrypting information 910 received from the encrypter 101 in the CIP header 954. An isochronous packet generator 107 adds an isochronous packet header 900, header CRC 901, and data CRC 903 to the isochronous payload 952 received from the CIP block generator 103, and outputs an isochronous packet. Since the content of the isochronous payload 952 is data that conforms to the AV protocol, the value of the tag 907 is set to 1. The key generator 106 sends the encryption key 109 to the receiving device 128 by communicating the asynchronous packet with the receiving device 128. The key generator 106 also outputs the encryption key 109 to the encrypter 101.

[0020]

A 1394 packet I/O controller 108 inputs and outputs isochronous and asynchronous packets between the 1394 bus 111 and sending device 110. More specifically, the 1394 packet I/O controller 108 outputs the isochronous packet received from the isochronous packet generator 107 and asynchronous packet received from the key generator 106 to the 1394 bus 111, and also outputs asynchronous packet received from the 1394 bus 111 to the key generator 106.

[0021]

A 1394 packet I/O controller 127 inputs and outputs isochronous and asynchronous packets between the 1394 bus 111 and receiving device 128. More specifically, the 1394 packet I/O controller 127 outputs the isochronous packet received from the 1394 bus 111 to a payload extractor 123, and outputs asynchronous packet received from the 1394 bus 111 to a key generator 125. The 1394 packet I/O controller 127 also outputs asynchronous packet received from the key generator 125 to the 1394 bus 111.

[0022]

The payload extractor 123 receives the isochronous packet, transmitted from the 1394 bus 111, from the 1394 packet I/O controller 127. When the value of the isochronous packet tag 907 is 1, the payload extractor 123 determines that an isochronous payload 952 contains data conforming to the AV protocol, and outputs the isochronous payload 952 to an actual data extractor 122. When received isochronous payload 952 contains the actual data 905, the actual data extractor 122 outputs the actual data 905 to a decrypter 121, after removing the CIP header 954 placed at the beginning of the isochronous payload 952. The actual data extractor 122 also outputs the source ID 906 and encrypting information 910 extracted from the CIP header 954 to the key generator 125. The encrypting information 910 is also output to the decrypter 121. The key generator 125 receives an encryption key 126 as a result of exchanging asynchronous packet with the sending device 110 via asynchronous communication outputs the encryption key 126 to the decrypter 121. When the value of the encrypting information 910 received from the actual data extractor 122 is 0, the decrypter 121 outputs the actual data 905 received from the actual data extractor 122 to an AV generator 120 as it is. When the value of the encrypting information 910 is 1, the decrypter 121 decrypts the actual data 905 using the encryption key 126 received from the key generator 125, and outputs decrypted actual data 905 to the AV generator 120.

[0023]

Figs. 3 show the command and response formats of the AKE commands (AKE: Authentication and Key Exchange) communicated between the key generators 106 and 125. These commands and responses belong to the AV/C Command Set, and are communicated between the sending device 110 and receiving device 128 using the asynchronous communication. By communicating these commands and responses, the sending device 110 and receiving device 128 exchange information required for the authentication of each other and encryption keys 109 and 126. The AKE commands consist of AKE control commands for requesting a specific operation to a target device and AKE status commands for querying the status and capabilities of the target device.

[0024]

Fig. 3(a) shows the format of the AKE status command. In the AKE status command, an operation code 208 indicates that this command is an AKE command. The value of the algorithm ID 200 is set at 0, with other values reserved for future extension.

[0025]

Fig. 3(b) shows the format of responses to the AKE status commands. This is a response sent back from the device receiving the AKE status command to the device issuing the AKE status command. There are multiple procedures for exchanging information for mutual authentication and transmission of encryption keys 109 and 126 between the sending device 110 and receiving device 128. In an algorithm field 201, the identifier for an information exchange procedure that the device returning an applicable response can execute is assigned in bits. In other words, the receiving device 128 exchanges several commands and responses with the sending device 110 after an encrypted TSP is detected in line with the aforementioned procedures and before receiving the encryption keys 109 and 126. There is plural information exchange procedure for communicating these commands and responses. The device sending back the response designates the executable information exchange procedure by setting 1 to an applicable bit in the algorithm field 201. Since the size of the algorithm field 201 is 16 bits, a maximum of 16 types of information exchange procedures can be indicated. The maximum data length 212 indicates the longest receivable data length in the form of bytes for exchanging AKE commands and responses.

[0026]

Fig. 3(c) shows the format of the AKE control commands. The algorithm field 201 in the AKE control commands shows an executed information exchange procedure when the value of the algorithm ID 200 is 0. Only one bit in the algorithm field 201 of the AKE control command and the response to AKE control commands is set at 1. A bit having the value 1 indicates the information exchange procedure being used. A label 202 is used for identifying correspondence between AKE control commands. For example, a certain information exchange procedure specifies that the device receiving an AKE control command needs to return a different AKE control command corresponding to the AKE control command received when the AKE control command is sent from one device to another. In this case, the label 202 inserted in the returned AKE control command has the same value as the label 202 inserted in the first AKE control command received, in order to clarify the correlation between both AKE control commands. In step No. 203, a serial number from 1 is given to each AKE control command in the sequence of communication in the information exchange procedure.

[0027]

A sub-function 299 takes the values shown in Table 1.

[0028]

Table 1

Sub-function	Value
Make-response	00_{16}
Verify-me	01_{16}
Create-key-information	10_{16}
Reconstruct-key	11_{16}
Exchange	20_{16}

[0029]

If the sub-function 299 is the make-response, this AKE control command challenges the authentication of the device receiving this command. Here, the data 207 contains authentication challenge data expressed as random numbers to authenticate the receiving device. The device receiving this command returns an AKE control command whose sub-function 299 is set to verify-me. When returning the AKE control command, the data stored in the data 207 is the authentication response data that is a result of a predetermined operation with respect to the authentication challenge data in the received data 207. The key information used for this operation is a key given only to an authorized device in advance. Whether the device executing the operation is an authorized device or not can be determined by checking the returned authentication response data.

[0030]

If the sub-function 299 is the create-key-information, this AKE control command requests the encryption key 109 to the device receiving this command. The device receiving this AKE control command returns the AKE control command whose sub-function 299 is set to reconstruct-key. At this point, the encrypted encryption key 109 is stored in the data 207 and returned.

[0031]

If the sub-function 299 is the exchange, this AKE control command requests the exchange of key information between devices sending and receiving the command. This key information is stored in the data 207 and transferred for indirect authentication between devices and the creation of a common key.

[0032]

Values of the sub-function other than those specified in Table 1 are reserved for future extension. The channel No. 204 indicates the channel number for isochronous communication between the sending device 110 and receiving device 128. This channel No. 204 is valid only when the sub-function 299 is set to the create-key-information or reconstruct-key. In other cases, this value is set to FF in hexadecimal format. Block No. 205 and total block No. 206 are used when data that should be handled by the AKE control command cannot be sent by one AKE command. In this case, applicable data is divided into blocks, and transferred in several transmissions. The total block No. 206 indicates the number of divided blocks in applicable data. The block No. 205 indicates the number of each block in the data 207. The data length 209 indicates the valid data length, as bytes, in the data 207. The data 207 is data exchanged by the AKE control command. The device receiving the AKE control command returns a response to that specific AKE control command. The format and value of the response are the same as those of the received AKE control command. The only detail that differs is that the response does not contain the data 207.

[0033]

Fig. 4 shows a time sequence example of AV/C commands that are exchanged between the sending device 110 and receiving device 128 before sending the encryption keys 109 and 126 from the sending device 110 to receiving device 128. First, operations of both devices before exchanging AV/C commands shown in Fig. 4 are briefly described.

[0034]

An initial condition is that non-encrypted TSP is sent from the sending device 110. At this time, the receiving device is not working. The TSP from the signal source 100 is inputted to the encrypter 101. Since the output command 105 is set to the normal output, the encrypter 101 outputs TSP as it is without encryption to the source packet generator 102, and registers the value 0 as the encrypting information 910. The source packet generator 102 adds 4-byte source packet header to the TSP received, and outputs it to the CIP block generator 103. The CIP block generator 103 adds 8-byte CIP header 954, and outputs it as isochronous payload 952 to the isochronous packet generator 107. Here, the encrypting information 910 contained in the CIP header 954 is 0 that is inputted from the encrypter 101. The isochronous packet generator 107 adds the isochronous packet header 900, header CRC 901, and data CRC 903 to the received isochronous payload 952 to create the isochronous packet. This isochronous packet is outputted to the 1394 bus 111. Since the applicable isochronous packet conforms to the AV protocol, the tag 907 in the isochronous packet header 900 is set to 1.

[0035]

When the TSP output from the signal source 100 is changed, that means AV information changes from that unprotected AV information to copy-protected AV information, the key generator 106 detects this change, and changes the output command 105 from the normal output to empty output. At the same time, the encryption key 109 for encrypting TSP is given to the encrypter 101.

[0036]

When the output command 105 is for empty output, the encrypter 101 outputs an empty signal to the source packet generator 102 every time it receives a TSP from the signal source 100, and registers the value 1 as the encrypting information 910. At receiving the empty signal from the encrypter 101, the source packet generator 102 transmits the received empty signal as it is to the CIP block generator 103 without adding the source packet header. When the CIP block generator 103 receives the empty signal, it outputs only the CIP header 954 to the isochronous packet generator 107. Here, the encrypting information 910 in the CIP header 954 uses the value 1 output from the encrypter 101. The isochronous packet generator 107 creates an isochronous packet as the isochronous payload 952 using the CIP header 954 received from the CIP block generator 103, and outputs it to the 1394 packet I/O controller 108. Since this isochronous packet conforms to the AV protocol, the value of the tag 907 is set to 1. The 1394 packet I/O controller 108 outputs received isochronous packet to the 1394 bus 111. This isochronous packet is continuously output, and the isochronous packet only containing the CIP header 954 in this isochronous payload 952 is continuously outputted to the 1394 bus 111. The receiving device 128 receiving this isochronous packet checks its tag 907 by the 1394 packet I/O controller 127, detects that the isochronous packet conforms to the AV protocol, and then outputs this isochronous packet to the payload extractor 123. The payload extractor 123 extracts the isochronous payload 952 from received isochronous packet, and outputs it to the actual data extractor 122. The actual data extractor 122 outputs the encrypting information 910 and source ID 906 in the CIP header 954 to the key generator 125. After the key generator 125 detects that the value of the encrypting information 910 is 1, the key generator 125 learns from the source ID 906 that device outputting the isochronous packet is the sending device 110. Then, the key generator 125 finally goes onto a process for requesting the encryption keys 109 and 126 using the A/C commands, as shown in Fig. 4.

[0037]

In Fig. 4, the AKE status command 300 is first sent from the receiving device 128 to sending device 110. This enables the receiving device 128 to query information exchange procedure that can be used by the sending device 110. Replying to this query, the sending device 110 returns the AKE response 301 to the receiving device 128. Information exchange procedure that the sending device 110 can execute is assigned in bits in the algorithm field 201 of the AKE response 301. This allows the receiving device 128 to learn which information exchange procedures can be executed by the sending device 110. For example, if the sending device 110 can execute the second and sixth information exchange procedures, binary indication in the algorithm field 201 of the AKE response 301 is 0000000000100010.

[0038]

The receiving device 128 receiving the AKE response 301 selects one optimal procedure from information exchange procedures that both sending device 110 and receiving device can execute. Then, AV/C commands are exchanged according to the selected exchange procedure.

Let's say the receiving device 128 can execute the second and eighth information exchange procedures. Then, the information exchange procedure that can be executed by both sending device 110 and receiving device 128 is only the second procedure. Accordingly, the rest of authentication and information exchange is executed using the second procedure. In the AKE control command in this procedure, the value of algorithm ID 200 is 0 and the value of the algorithm field 201 is 0000000000000010 in binary indication. The information exchange procedure specifies not only the sequence of exchanging a range of AKE control commands but also a format and processing method of the data 207 sent by each AKE control command.

[0039]

In accordance with the second information exchange procedure, the key generator 125 sends the make-response command 302 to the sending device 110. In the data 207 of this make-response command 302, two random numbers RRa and RRb generated by the key generator 125 are encrypted, and the algorithm field 201 contains identification information indicating the use of the second procedure. The key used for encryption is a common secret key given to both authorized sending device and receiving device in advance. The key generator 106 receiving the make-response command 302 checks the algorithm field 201 of the received make-response command 302, and learns to use the second procedure for the rest of the authentication and information exchange. Since the key generator 106 can execute the second procedure, the data 207 of the make-response command 302 sent in accordance with this second procedure are known to contain two random numbers encrypted by this secret key. After taking out two random numbers RRa and RRb from the data 207 using this secret key, the key generator 106 returns a response 303 to inform that a response can be generated. Then, the key generator 106 stores one of the random numbers RRa taken out in the data 207, and sends the verify-me command 304 to the receiving device 128. This is the response requested by the previous make-response command 302. Hereafter, algorithm area 201 of each AKE command that is exchanged between the sending device 110 and receiving device 128 always contains the identification information indicating the second procedure.

[0040]

The key generator 125 receiving the verify-me command 304 confirms that RRa in the data 207 conforms to the random number RRa generated by itself, and then returns a response 305 to the verify-me command 304 to inform that verification has completed successfully. The key generator 125 then finally authenticates that the sending device 110 is an authorized sending device.

[0041]

The sending device 110 then use the make-response command 306 and verify-me command 308 in accordance with the procedures after the make-response command 302 described above to confirm that the receiving device 128 is an authorized receiving device. However, the random number used here is RTa and RTb, and the random number sent back by the verify-me command 308 is RTb.

[0042]

Now, both sending device 110 and receiving device 128 know the random numbers RRb and RTb. It is confirmed that both are authorized devices. The key generator 106 and key generator 125 separately generates a temporary key (not illustrated) from RRb and RTb using a common operation method specified by the second procedure. These temporary keys are a common key existing only between the sending device 110 and receiving device 128.

[0043]

Next, the key generator 125 sends the create-key-information command 310 to the sending device 110. A channel number of the isochronous packet that the receiving device 128 is currently receiving is stored in the channel No. 204 of the create-key-information command 310. The key generator 106 receiving this create-key-information command 310 encrypts the encryption key 109 to be used for encrypting TSP with the aforementioned temporary key, and then returns a response 311 to inform that the create-key-information command 310 has completed successfully. Then, the key generator 106 sends the reconstruct-key command 312 that stores the encryption key 109 encrypted by the temporary key in its data 207 to the receiving device 128. The key generator 125 uses the temporary key to decrypt the data 207 of the reconstruct-key command 312 received, and obtains the encryption key 126. Then, the key generator 125 returns a response 313 to inform that the reconstruct-key command 312 has completed successfully. Since the encryption keys 109 and 126 are encrypted and decrypted using the same temporary key, they are the same keys. The encryption key

126 is outputted from the key generator 125 to the decrypter 121.

[0044]

The key generator 106 that has sent the reconstruct key command 312 outputs the command 105 for encrypted output to the encrypter 101. The encrypter 101 receiving this command encrypts TSP received from the signal source 100 by the encryption key 109, and starts to output it to the source packet generator 102. This enables the sending device 110 to send the isochronous packet containing TSP encrypted by the encryption key 109 in its isochronous payload 952 on the 1394 bus 111. This isochronous packet received by the receiving device 128 is decrypted by the decrypter 121 using the encryption key 126 as described above, and outputs the decrypted packet to the AV generator 120.

[0045]

In the above series of AKE control commands, each set of the make-response command 302 and verify-me command 304; make-response command 306 and verify-me command 308; and create-key-information command 310 and reconstruct-key command 312, respectively has the same label 202. The make-response command 302, verify-me command 304, make-response command 306, verify-me command 308, create-key-information command 310, and reconstruct- key command 312 also have values 1, 2, 3, 4, 5, and 6 in the step No. 203 respectively.

[0046]

If the actual data in the isochronous packet output from the sending device 110 changes from encrypted actual data to non-encrypted actual data, the decrypter 121 detects the change in the encrypting information 910 and stops decryption. The data received from the actual data extractor 122 is outputted as it is to the AV generator 120.

[0047]

If a bus reset occurs in the 1394 bus 111 after the aforementioned processes shown in Fig. 4 starts, the procedures after and make-response command 302 need to be repeated.

[0048]

As described above, in the preferred embodiment of the present invention, the sending device sends encrypting information that indicates the encryption status of the actual data in the isochronous packet together with the actual data. This enables the receiving device receiving the isochronous packet to make a request to the sending device for an encryption key for decrypting the actual data if the receiving device detects, by checking the encrypting information in the isochronous packet, that the actual data is encrypted. The sending device receiving the request then gives the encryption key to the receiving device. Accordingly, the data transfer method of the present invention offers extremely simple procedures for giving the encryption key for decryption from the sending device to receiving device.

[0049]

The isochronous packet transferred via isochronous communication contains i)encrypting information indicating the encryption status of the actual data and

ii) actual data, but only the actual data is encrypted for data transfer. This makes possible a data transfer method that has no risk of erroneous operation when a conventional receiving device receives encrypted actual data while maintaining compatibility with the conventional data transfer method.

[0050]

Furthermore, in the preferred embodiment of the present invention, the CIP header remains non-encrypted for transfer even if encryption of synchronous data starts while the receiving device is continuously receiving synchronous data sent by the sending device. This enables a data transfer method that eliminates the possibility of erroneous operation of the receiving device receiving the data.

[0051]

In the preferred embodiment, once encryption by the encryption key starts, actual data in all transfer units is encrypted and sent. However, it is not necessary to encrypt all units of data to be transferred. For example, even if both encrypted transfer units and non-encrypted transfer units are sent alternately, the receiving device can correctly decrypt the data because encrypting information is included in the CIP header, thus achieving the same effect. In addition, it is apparent that the same effect is also achievable even if the receiving device specifies a percentage of encrypted transfer units to the sending device. The size of the MPEG source packet is 192 bytes, with more than one source packet stored in one isochronous payload in the case of the high data rate transfer of MPEG (12 Mbps minimum). Naturally, however, it is not possible to have both encrypted source packet and non-encrypted source packet in the same isochronous payload.

[0052]

In the preferred embodiment, all actual data is encrypted using the encryption key. However, it is not necessary to encrypt all pieces of data. For example, the same effect is achievable by encrypting the first half of the actual data, or encrypting the first and third quarters of the actual data. In this case, the receiving device can decrypt appropriately, if, when sending the data, information is inserted to indicate encrypted portions and their percentage in the CIP header. The same effect is also achievable by inserting in the CIP header encrypting information announcing whether the actual data is encrypted or not. The receiving device queries the sending device via asynchronous communication about which part of the actual data is encrypted and to what level, when the receiving device detects encryption by checking the CIP header. The same effect is also achievable in this case even when the receiving device specifies the encryption area and percentage to the sending device via asynchronous communication. If only the confidential portion in the actual data is encrypted, the burden for encryption and decryption is reduced, and at the same time, a sufficient effect of encryption may be achieved.

[0053]

In the preferred embodiment, the isochronous packet containing only the CIP header without actual data is transferred until the completion of mutual authentication between the sending and receiving devices. However, the same effect is achievable even when an isochronous packet containing encrypted actual data is outputted from the start, and not the isochronous packet containing only the CIP header.

[0054]

In the preferred embodiment, procedures for transferring the AKE control commands between sending and receiving devices are determined by mutual negotiation. However, if the receiving device features only one executable procedure, the same effect is achievable by starting to transfer commands immediately, without executing this negotiation procedure, using only the executable procedure. In this case, it may be preferable to specify in advance a basic minimum of executable procedures for all authorized devices.

[0055]

In this preferred embodiment, direct authentication is implemented between the sending and receiving devices, following which decrypting information is transferred using a secret key. However, the means for transferring authentication and decrypting information is not limited to this procedure. For example, a public key may be used for mutual indirect authentication and the creation of a temporary key. Decrypting information may then be transmitted using this temporary key. Such procedures are briefly described below.

[0056]

The sending and receiving device stores the key information necessary for mutual indirect authentication in the data 207 of the AKE control command, and send this information to each other in line with a procedure determined by mutual negotiation. Here, the sub-function 299 is set to the exchange. This enables both sending and receiving devices to share the same temporary key if they are both authorized devices. Decrypting information is then transferred using the create-key-information command and reconstruct-key command in accordance with the same procedures as those described in the preferred embodiment.

[0057]

[ADVANTAGE OF THE INVENTION]

As described above, the present invention has the significant effect of realizing a data transfer method using extremely simple procedures for passing key information for decryption from the sending device to the receiving device. Encryption identification information indicating the encryption status of actual data in synchronous data is sent together with actual data. The receiving device receiving the synchronous data checks the encryption identification information in the synchronous data, and if it detects that the actual data is encrypted, the receiving device requests the sending device for decrypting information for decrypting the encrypted data. The sending device receiving this request gives the decrypting information to the receiving device.

[0058]

The present invention has another significant effect of realizing a data transfer method which eliminates the possibility of erroneous operation of the receiving device even if conventional receiving device receives encrypted synchronous data, while maintaining compatibility with a conventional data transfer method. Synchronous data transferred through synchronous communication contains i) encryption identification information indicating encryption status of the actual data and ii) actual data, but only the actual data is encrypted for data transfer.

[0059]

The present invention has still another significant effect of realizing a data transfer method that eliminates the possibility of erroneous operation of the receiving device even if encryption of synchronous data starts while the receiving device continuously receives synchronous data sent from the sending device. Synchronous data transferred through synchronous communication contains i) encryption identification information indicating encryption status of the actual data and ii) actual data, but only the actual data is encrypted for data transfer. This enables to transfer the CIP header as it is without being encrypted.

[0060]

The present invention has still another significant effect of realizing a data transfer method that always executes the most suitable procedure even when new and conventional devices share the same network. A procedure for transferring and receiving both authentication and decrypting information with good future extendibility are achievable by selecting a procedure for providing authentication information and decrypting information exchanged between the sending and receiving devices by negotiation between the sending and receiving devices. In other words, when a new authentication method or decrypting information become available in the future, the most suitable procedure remains selectable by negotiation between devices. That is possible, even if a device which can use the new procedure and a device that can use only conventional procedures share the same network, as long as the new device is back-compatible with older procedures.

[0061]

The present invention allows the relative proportion of encrypted actual data and non-encrypted actual data to be varied. Accordingly, even if the receiving device has no exclusive hardware for decrypting the encrypted actual data, software can be used instead. More specifically, even if the receiving device has no special hardware for decryption like PC, decryption with the software is possible by reducing the proportion of encrypted actual data.

[0062]

Further, it is possible to vary the portion and the proportion of encrypted actual data. Accordingly, even if the receiving device has no exclusive hardware for decrypting the encrypted actual data, software can be used instead. More specifically, even if the receiving device has no special hardware for decryption like PC, decryption with the software is possible by reducing the proportion of encrypted actual data.

[0063]

The present invention has still another significant effect of realizing a data transfer method that uses the limited bus transfer band efficiently and significantly reduces the risk of unauthorized device receiving readable data. Unless the sending and receiving devices mutually authenticate that both are authorized devices, isochronous packets without actual data are outputted.

[BRIEF DESCRIPTION OF THE DRAWINGS]

(Fig. 1)

Schematic view of a format of a CIP header in accordance with a preferred embodiment of the present invention

(Fig. 2)

Block diagram for depicting relationship between sending and receiving devices in accordance with the preferred embodiment of the present invention

(Fig. 3)

(a) Format of AKE status command in accordance with the preferred embodiment of the present invention

(b) Format of AKE response to the AKE status command in accordance with the preferred embodiment of the present invention

(c) Format of AKE control command in accordance with the preferred embodiment of the present invention

(Fig. 4)

Schematic view illustrating procedures for transmitting an asynchronous packet between sending and receiving devices in accordance with the preferred embodiment of the present invention

(Fig. 5)

Format of isochronous packet in a data transfer method of the prior art

[DESCRIPTION OF THE MARKS]

101. Encrypter
102. Source packet generator
103. CIP block generator
106,125. Key generator
107. Isochronous packet generator
108,127. 1394 Packet I/O controller
110. Sending device
111. IEEE 1394 bus
121. Decrypter
122. Actual data extractor
123. Payload extractor
128. Receiving device
901. Header CRC
902,952. Isochronous payload
903. Data CRC
904,954. CIP header
905. Actual data
906. Source ID
907. Tag
910. Encrypting information

[Name of the document] Drawing

Fig. 1

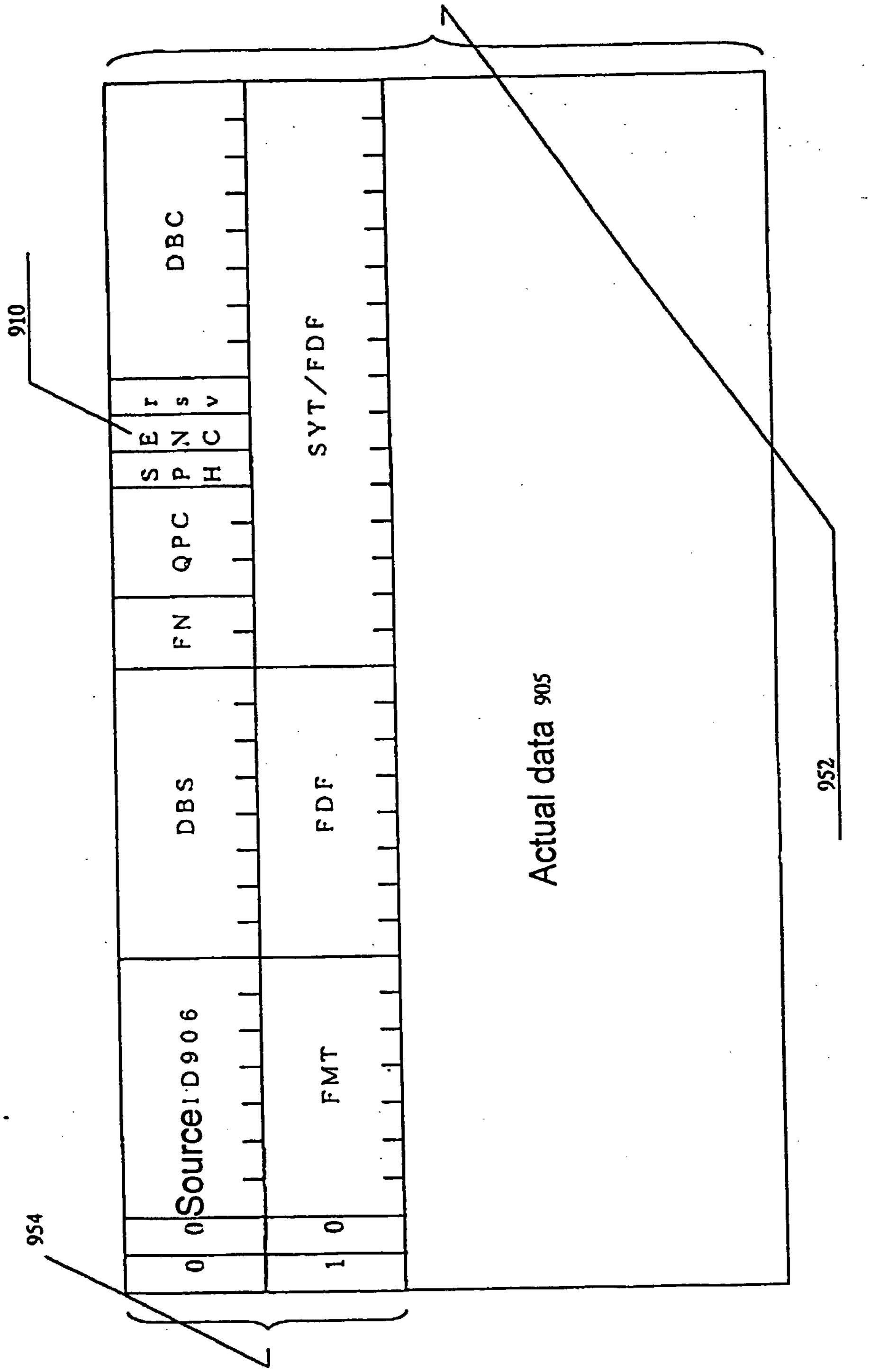
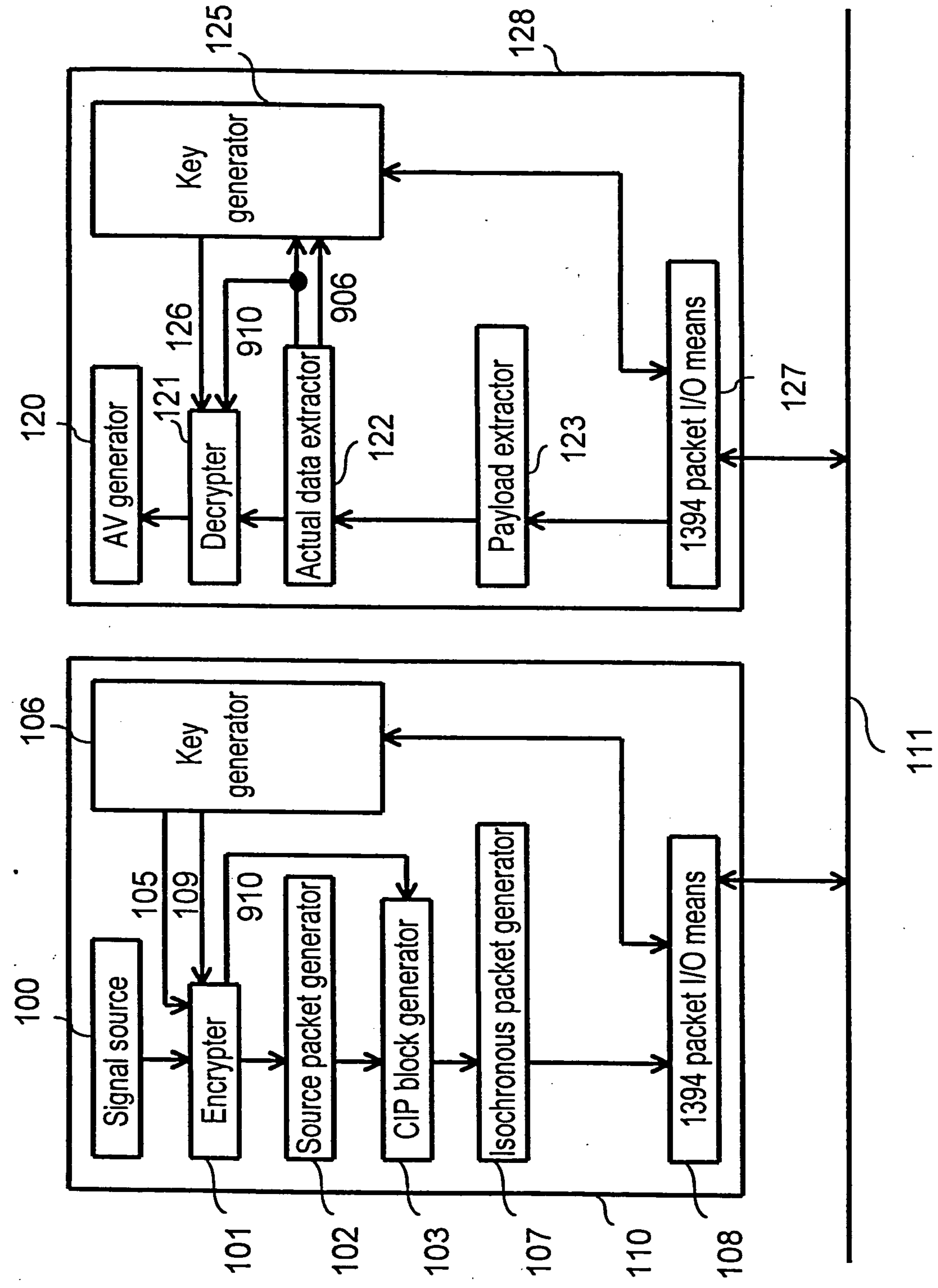
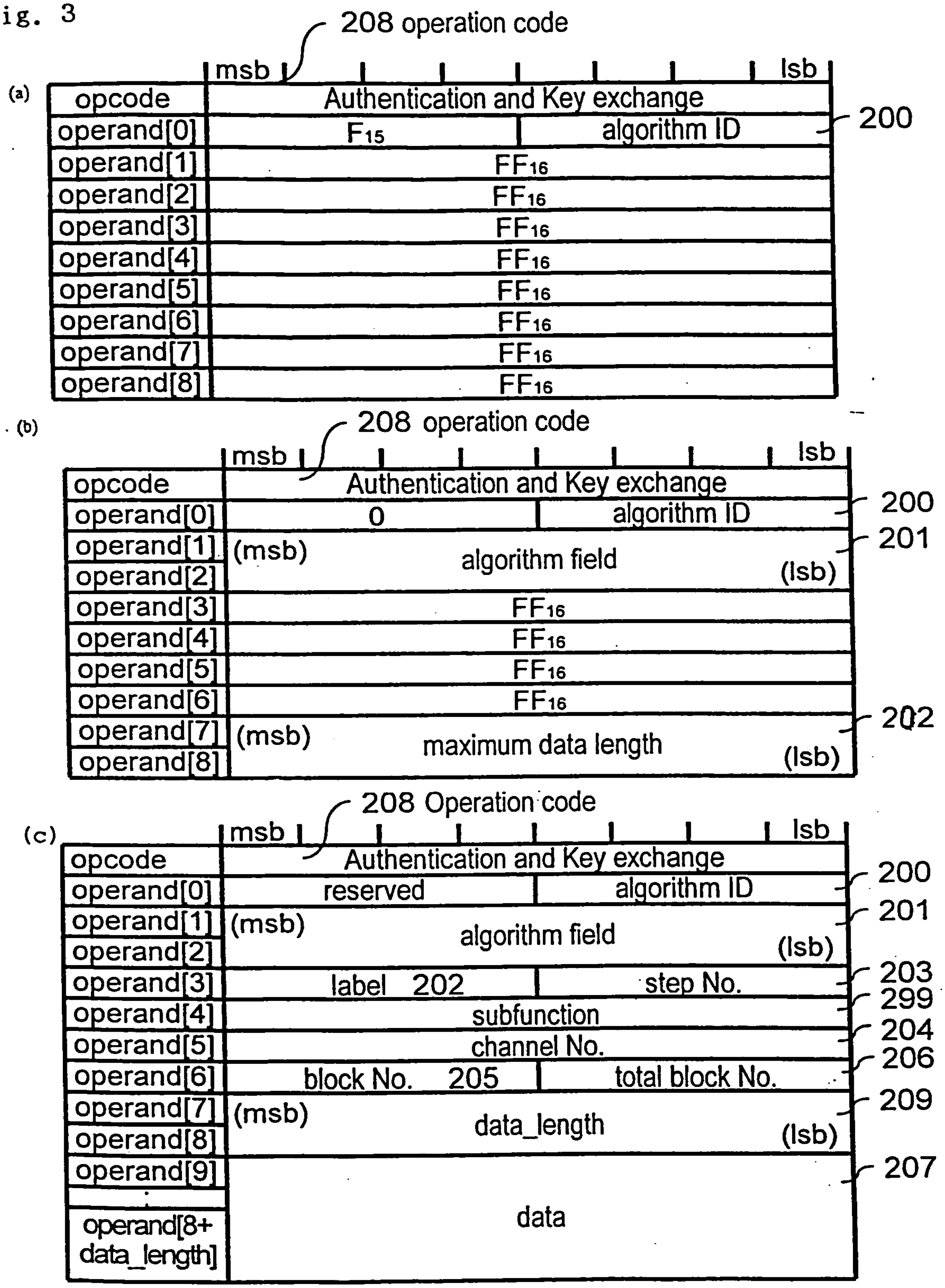


Fig. 2



125
Key generator
128
126
910
906
120
AV generator
121
Decrypter
Actual data extractor
122
Payload extractor
123
1394 packet I/O means
127
106
Key generator
105
109
910
100
Signal source
Encrypter
Source packet generator
CIP block generator
Isochronous packet generator
1394 packet I/O means
111
101
102
103
107
110
108

Fig. 3



(a)

	msb	lsb
opcode	Authentication and Key exchange	
operand[0]	F_{15}	algorithm ID
operand[1]	FF_{16}	
operand[2]	FF_{16}	
operand[3]	FF_{16}	
operand[4]	FF_{16}	
operand[5]	FF_{16}	
operand[6]	FF_{16}	
operand[7]	FF_{16}	
operand[8]	FF_{16}	

(b)

	msb	lsb
opcode	Authentication and Key exchange	
operand[0]	0	algorithm ID
operand[1]	(msb) algorithm field (lsb)	
operand[2]		
operand[3]	FF_{16}	
operand[4]	FF_{16}	
operand[5]	FF_{16}	
operand[6]	FF_{16}	
operand[7]	(msb) maximum data length (lsb)	
operand[8]		

(c)

	msb	lsb
opcode	Authentication and Key exchange	
operand[0]	reserved	algorithm ID
operand[1]	(msb) algorithm field (lsb)	
operand[2]		
operand[3]	label 202	step No.
operand[4]	subfunction	
operand[5]	channel No.	
operand[6]	block No. 205	total block No.
operand[7]	(msb) data_length (lsb)	
operand[8]		
operand[9]	data	
:		
operand[8+ data_length]		

Fig. 4

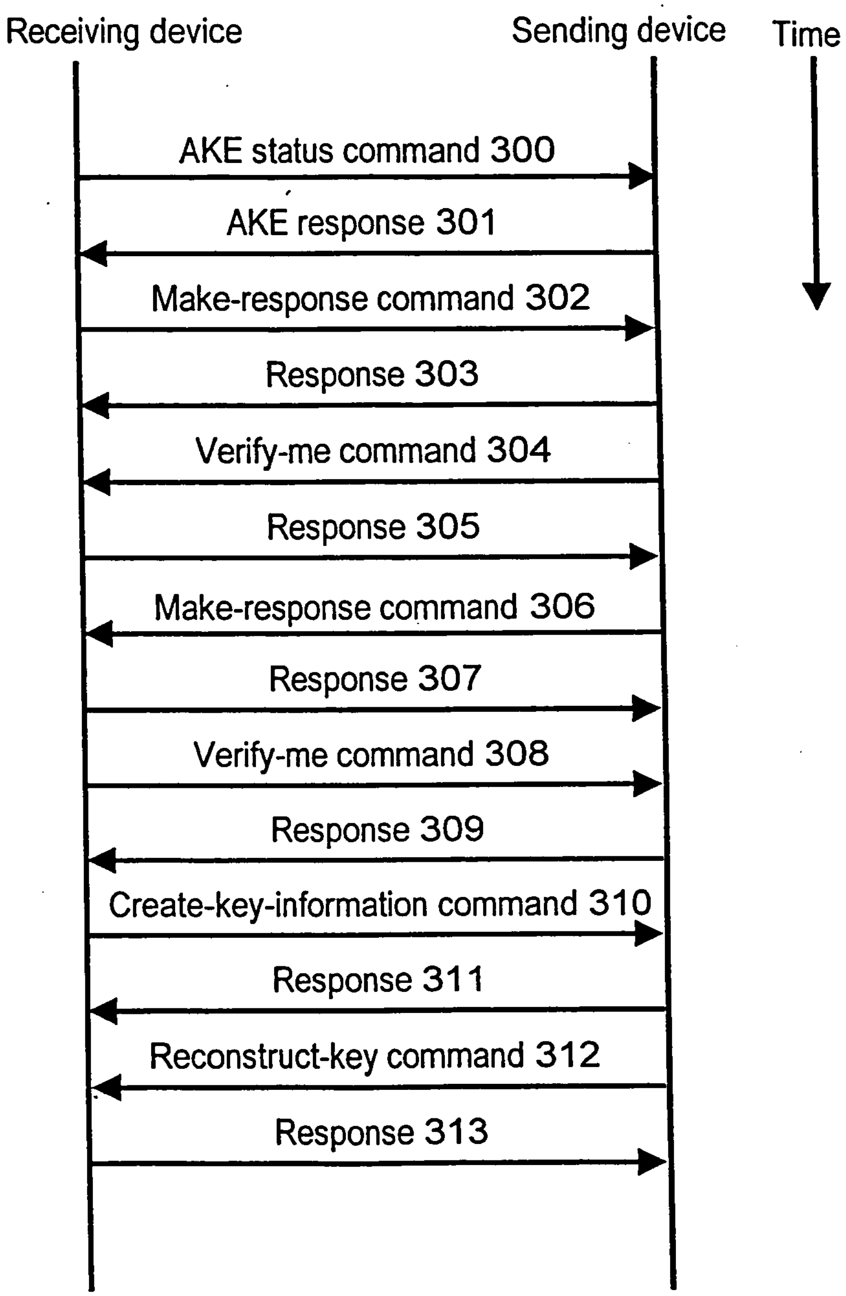
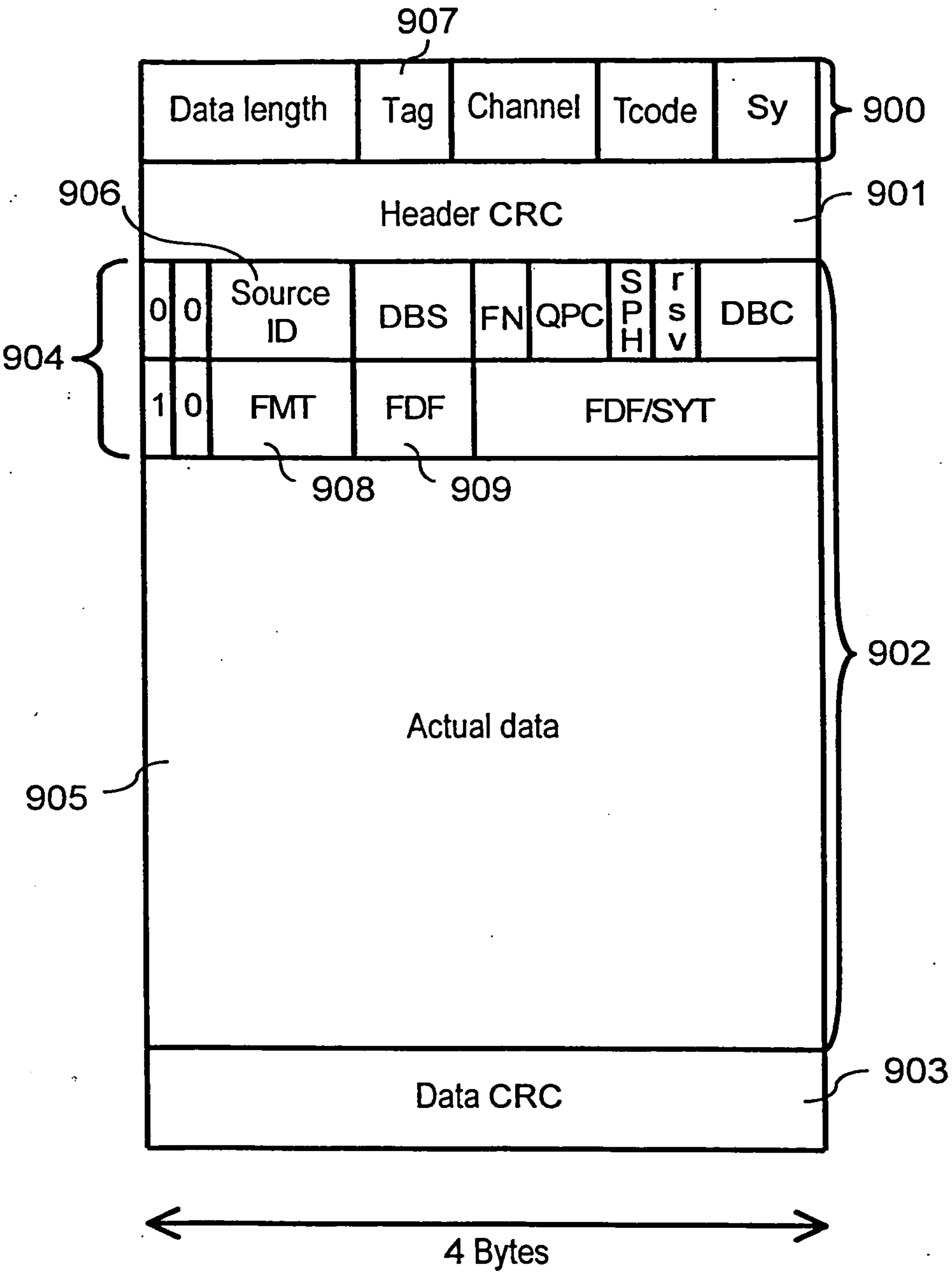

Receiving device
Sending device
Time
AKE status command 300
AKE response 301
Make-response command 302
Response 303
Verify-me command 304
Response 305
Make-response command 306
Response 307
Verify-me command 308
Response 309
Create-key-information command 310
Response 311
Reconstruct-key command 312
Response 313

Fig. 5


907
Data length
Tag
Channel
Tcode
Sy
900
906
Header CRC
901
904
0
0
Source ID
DBS
FN
QPC
S P H
r s v
DBC
1
0
FMT
FDF
FDF/SYT
908
909
902
Actual data
905
Data CRC
903
4 Bytes

[NAME OF THE DOCUMEMT] ABSTRACT

[OBJECT]

The present invention is to offer a data transfer method that eliminates erroneous operation of conventional devices not supporting encryption when copy-protected AV information is encrypted and sent on an IEEE 1394 bus.

[MEANS TO SOLVE THE PROBLEM]

Only actual data 905 is encrypted, and header 954 is sent without being encrypted.

Encryption identification information 910 indicating the encryption status of the actual data 905 is included in the header 954.

This satisfies the conventional standard, and erroneous operation is prevented when the conventional device receives such encrypted packet.

[SELECTED DRAWING] Fig. 1